

# Efficient simulations of tubulin-driven axonal growth

Stefan Diehl · Erik Henningsson · Anders Heyden

Received: 29 January 2016 / Revised: 14 March 2016 / Accepted: 5 April 2016  
The final publication is available at Springer via <http://dx.doi.org/10.1007/s10827-016-0604-x>

**Abstract** This work concerns efficient and reliable numerical simulations of the dynamic behaviour of a moving-boundary model for tubulin-driven axonal growth. The model is nonlinear and consists of a coupled set of a partial differential equation (PDE) and two ordinary differential equations. The PDE is defined on a computational domain with a moving boundary, which is part of the solution. Numerical simulations based on standard explicit time-stepping methods are too time consuming due to the small time steps required for numerical stability. On the other hand standard implicit schemes are too complex due to the nonlinear equations that needs to be solved in each step. Instead, we propose to use the Peaceman–Rachford splitting scheme combined with temporal and spatial scalings of the model. Simulations based on this scheme have shown to be efficient, accurate, and reliable which makes it possible to evaluate the model, e.g. its dependency on biological and physical model parameters. These evaluations show among other things that the initial axon growth is very fast, that the active transport is the dominant reason over diffusion for the growth velocity, and that the polymerization rate in the growth cone does not affect the final axon length.

**Keywords** Neurite elongation · Partial differential equation · Numerical simulation · Peaceman–Rachford splitting scheme · Polymerization · Microtubule cytoskeleton

## 1 Introduction

We are interested in the modelling of axonal elongation, or growth, from the stage when one of the developed neurites of the cell body (soma) of a neuron, begins to grow fast leaving the others behind. The growth can continue for a long time although with decreasing speed, and axons may also shrink. The main protein building material of the cytoskeleton consists of tubulin dimers, which are produced in the soma and transported to the tip of the axon, the growth cone, in which polymerization of the dimers to microtubules occurs. This simplified description of the mechanism of the one-dimensional elongation of the axon has been the focus of both experimental and theoretical works. For example, the purpose of theoretical work can be to investigate fundamental questions like the role of advection and diffusion for the transport of tubulin in long axons without performing tedious experiments. For references on axonal growth and different types of modelling of the behaviour of the axon and its growth cone, we refer to the review papers by Kiddie et al (2005); Graham and van Ooyen (2006); Miller and Heidemann (2008); van Ooyen (2011); Suter and Miller (2011) and the references therein.

The dynamic behaviour of a phenomenon is commonly modelled by differential equations. When an entity, like the concentration of tubulin along the axon, depends both on time and space, the conservation of mass leads to one or several partial differential equations (PDEs) (Smith and Simmons, 2001; McLean and Graham, 2004; Graham et al, 2006; Sadegh Zadeh and Shah, 2010; García et al, 2012; Diehl et al, 2014). Tubulin is in fact present in different states within an axon: motor protein-bound tubulin and free tubulin. Smith and Simmons (2001) presented and analyzed an accurate model of bidirectional transport by motor proteins and free tubulin. These three states are modelled by

E. Henningsson was supported by the Swedish Research Council under grant no. 621-2011-5588.

S. Diehl · E. Henningsson · A. Heyden  
Centre for Mathematical Sciences, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden  
E-mail: [diehl@maths.lth.se](mailto:diehl@maths.lth.se) (S. Diehl), [erikh@maths.lth.se](mailto:erikh@maths.lth.se) (E. Henningsson, corresponding author), [heyden@maths.lth.se](mailto:heyden@maths.lth.se) (A. Heyden).

three PDEs, two advection equations for the anterograde (outward from the cell body to the growth cone) and retrograde (inward) active transports, and one diffusion equation for the movement of free tubulin. The equations are coupled via reaction terms, or rather binding/detachment terms, which model the movements of substance between the free state and either of the actively moving-cargo states. Their model was successfully calibrated to published experimental data by Sadegh Zadeh and Shah (2010).

In their publication, Smith and Simmons (2001) also presented a simplified model of their three linear PDEs consisting of a single advection-diffusion PDE with only two lumped model parameters; an effective drift velocity and an effective drift diffusion constant; see Smith and Simmons (2001, Formulas (4a)–(4b)). It is such an equation, with an additional sink term modelling the degradation of tubulin, that was used by McLean and Graham (2004); Diehl et al (2014) and which we use in the present work.

Since the axon grows, the spatial interval where the tubulin concentration is defined varies in length and this leads to a moving-boundary problem. Such a model was presented by Diehl et al (2014) consisting of a PDE defined on an interval with moving boundary coupled to two ordinary differential equations (ODEs). One ODE models the speed of the axon growth, which depends on the assembly (and disassembly) processes in the growth cone. This ODE was formulated based on experimental evidence from literature. The assembly process depends on the available concentration of free tubulin in the growth cone, which in turn can be modelled by another ODE for the mass balance of tubulin in the cone. Since this mass balance contains the flux of tubulin along the axon into the growth cone, the latter ODE is coupled to the PDE. Hence, even for very simplified assumptions, the mathematical model becomes complicated. All steady-state solutions were presented in Diehl et al (2014) and their dependencies on the values of the biological and physical parameters were investigated. We refer to that publication for a detailed comparison with previously published models of axonal growth, in particular, by McLean and Graham (2004); McLean et al (2004); Graham et al (2006); McLean and Graham (2006), since our model can be seen as an extension of theirs.

It was possible to investigate the dependence on the model parameters of the steady-state solutions by means of explicit formulas (Diehl et al, 2014). Furthermore, the stability of each steady state was investigated by numerical simulations. If a mathematical solution is not stable under disturbances, it is not physically or biologically relevant and cannot appear in reality. Thus, while it was possible to describe all steady-state solutions with explicit formulas, numerical simulation had to be used for dynamic solutions. As was already noticed by McLean and Graham (2004); McLean et al (2004); Graham et al (2006); Diehl et al (2014), it is not

straightforward to perform reliable numerical simulations in reasonable CPU times. The moving boundary can be transformed to a stationary one; however, if one wants to simulate the outgrowth of an axon from a very small initial length to its final one, several magnitudes larger, simulations can take months of CPU time to perform unless a tailored numerical method is used.

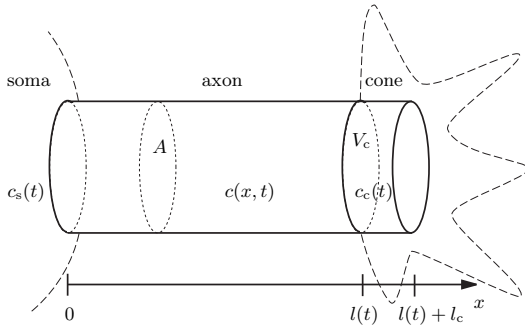
It is the main purpose of this article to present an efficient numerical scheme that can be used for the simulation of the dynamic behaviour of axonal growth. We also demonstrate the difficulties of using a standard method. Moreover, we present simulations of the dynamic behaviour of both growth and shrinkage for variations in the parameters. These simulations give a deeper insight in the parameters' influence on axonal growth and complement the information from the steady-state solutions presented in Diehl et al (2014).

The efficient numerical scheme presented is obtained by transforming the model in both space and time, applying a standard second-order spatial discretization, and using the Peaceman-Rachford splitting time-discretization (Douglas, 1955; Peaceman and Rachford, 1955; Hundsdorfer and Verwer, 2003; Hansen and Henningsson, 2013). Numerical investigations for both the short and long time behaviour indicate the convergence of the numerical solutions to those of the differential equations, although no proof of convergence is provided. Furthermore, simulations converge to exact steady-state solutions when the input soma concentration is constant.

The model equations are reviewed in Sec. 2 together with the model parameters. In Sec. 3, the transformations of the equations in both space and time are given and these are used for the numerical methods presented in Sec. 4. Then Sec. 5 contains several simulations performed partly to investigate the properties of the numerical methods as such, and partly to investigate the dynamical properties of the axonal-growth model. Conclusions are found in Sec. 6.

## 2 The model

An idealized one-dimensional axon is shown in Fig. 1. The axon length  $l(t)$  [m] at time  $t$  [s] is measured from the soma at  $x = 0$  to the growth cone. The effective cross-sectional area  $A$  [m<sup>2</sup>] of the axon through which tubulin is transported is assumed to be constant. Tubulin is produced in the soma, which is assumed to have the known concentration  $c_s(t)$ . This function is the driving input to the model. The unknown concentration of tubulin along the axon is denoted by  $c(x, t)$  [mol/m<sup>3</sup>] and in the growth cone by  $c_c(t)$ . Along the axon, both the motor protein-bound and the free tubulin are included in  $c(x, t)$ . No tubulin is produced along the axon, but degradation occurs at the constant rate  $g$  [1/s]. The active transport by motor proteins is assumed to occur at the



**Fig. 1** Schematic illustration of a growing axon.

the constant velocity  $a$  [m/s] and the diffusion of free tubulin is modelled by Fick's law with a constant diffusion coefficient  $D$  [m<sup>2</sup>/s]. The growth cone has the volume  $V_c$  [m<sup>3</sup>]. It turns out that the equations contain the ratio  $l_c := V_c/A$ , which we therefore interpret as a length parameter characterizing the size of the growth cone. In the cone, consumption of tubulin occurs by degradation at the constant rate  $g$  [1/s] and by assembly of dimers to microtubules, which elongates the axon at a constant rate  $\tilde{r}_g$  [1/s], i.e.,  $\tilde{r}_g$  is the reaction rate of polymerization of guanosine triphosphate (GTP) bound tubulin dimers to microtubule bound guanosine diphosphate (GDP). We let  $A_g$  [m<sup>2</sup>] denote the constant effective area of polymerization growth and  $\rho$  [mol/m<sup>3</sup>] the density of the assembled microtubules (the cytoskeleton). Additionally, we assume that the assembled microtubules in the growth cone may disassemble at the constant rate  $\tilde{s}_g$  [1/s]. All biological and physical constants are assumed to be positive.

The model equations are the following:

$$\begin{cases} \frac{\partial c}{\partial t} + a \frac{\partial c}{\partial x} - D \frac{\partial^2 c}{\partial x^2} = -gc, & 0 < x < l(t), t > 0, & (1a) \\ \frac{dc_c}{dt} = \frac{(a - gl_c)c_c - D}{l_c} \frac{dc_x^-}{dx} - \frac{(r_g c_c + \tilde{r}_g l_c)(c_c - c_c^\infty)}{l_c}, & t > 0, & (1b) \\ \frac{dl}{dt} = r_g(c_c - c_c^\infty), & t > 0, & (1c) \\ c(0, t) = c_s(t), & t \geq 0, & (1d) \\ c(l(t), t) = c_c(t), & t > 0, & (1e) \\ c(x, 0) = c^0(x), & 0 \leq x < l(0) = l^0, & (1f) \\ c_c(0) = c^0(l^0). & & (1g) \end{cases}$$

Equation (1a) models the tubulin concentration along the axon, influenced by advection, diffusion and degradation. Here we have used the common assumption that the flux [mol/(m<sup>2</sup>s)] of tubulin is

$$F(c, c_x) = ac - Dc_x, \quad (2)$$

where  $c_x := \partial c / \partial x$ . The conservation of tubulin in the growth cone is described by (1b), which we derive below

after motivating Equation (1c). The latter equation states that the growth velocity due to (net) polymerization is an affine function of the available concentration  $c_c$  in the cone. Since  $c_c(t) = c_c^\infty$  is equivalent to  $l'(t) = 0$ , the constant  $c_c^\infty$ , appearing both in (1b) and (1c), is the steady-state concentration at which the processes of assembly and disassembly are equally fast. The background of Equation (1c) is partly the assumption that the assembly of tubulin dimers is assumed to be proportional to the amount of tubulin in the cone  $V_c c_c(t)$  with the reaction rate  $\tilde{r}_g$  as the proportionality constant, and partly that the disassembly occurs at the rate  $\tilde{s}_g$  and is proportional to the amount of already assembled microtubules,  $\rho A_g \kappa l_c$ , where  $\kappa > 0$  is a dimensionless constant such that  $\kappa l_c$  is the length of the assembled microtubules that may undergo disassembly. Hence, this disassembly does not depend on the concentration of free tubulin, an assumption in accordance with experiments presented by Walker et al (1988). This leads to the equation

$$\underbrace{\frac{d(\rho A_g l)}{dt}}_{\text{mass increase per unit time}} = \underbrace{\tilde{r}_g V_c c_c}_{\text{assembly}} - \underbrace{\tilde{s}_g \rho A_g \kappa l_c}_{\text{disassembly}}. \quad (3)$$

which can be written as

$$\frac{dl}{dt} = r_g c_c - s_g \quad \text{with} \quad r_g := \frac{\tilde{r}_g V_c}{\rho A_g} \quad \text{and} \quad s_g := \tilde{s}_g \kappa l_c, \quad (4)$$

Here  $s_g$  is the maximum speed of shrinkage, which occurs when  $c_c = 0$ . The lumped parameter  $r_g$  is a concentration-rate constant. To convert (4) to (1c), we define the constant

$$c_c^\infty := \frac{s_g}{r_g} = \frac{\tilde{s}_g \rho A_g \kappa l_c}{\tilde{r}_g V_c} = \frac{\tilde{s}_g \rho A_g \kappa}{\tilde{r}_g A}. \quad (5)$$

Equation (1b) originates from the conservation of mass of tubulin in the growth cone  $c_c$ :

$$\underbrace{\frac{d(V_c c_c)}{dt}}_{\text{mass increase per unit time}} = \underbrace{A(ac^- - Dc_x^- - l'c^-)}_{\text{flux in}} - \underbrace{gV_c c_c}_{\text{degradation}} - \underbrace{\tilde{r}_g V_c c_c}_{\text{assembly}} + \underbrace{\tilde{s}_g \rho A_g \kappa l_c}_{\text{disassembly}}. \quad (6)$$

The assembly and disassembly terms here are the same as in (3), however, with opposite signs. The flux [mol/s] of tubulin into the growth cone is the product of  $A$ , the concentration just to the left of  $x = l(t)$ , which is

$$c^- = c^-(t) := c(l(t)^-, t) = \lim_{\varepsilon \searrow 0} c(l(t) - \varepsilon, t), \quad (7)$$

and the net velocity of tubulin across  $x = l(t)$ . The velocity due to advection and diffusion is  $F(c^-, c_x^-)/c^-$  relative the

axon, where  $F$  is defined in (2) and  $c_x^-$  is defined in the similar way as (7). Since the axon is elongated with the speed  $l'(t)$ , the net flux across the moving boundary  $x = l(t)$  is

$$\begin{aligned} c^- \left( \frac{F(c^-, c_x^-)}{c^-} - l' \right) &= F(c^-, c_x^-) - l' c^- \\ &= a c^- - D c_x^- - l' c^-, \end{aligned}$$

which explains the flux term of (6). That equation can now be rewritten by dividing by  $V_c = A l_c$ , and using (1c) and (5) to obtain (1b).

Initial data are denoted by the super index 0. In the derivations of the model equations (1a)–(1g), we have used the natural assumption that the concentration of tubulin is continuous in space and time.

The parameter values used are shown in Table 1. The nominal values are extracted carefully from the biological literature and we refer to Diehl et al (2014) for references and explanations on how the parameter values were found. In particular, the nominal values for  $r_g$  and  $s_g$  were calculated from the experiments reported by Walker et al (1988). The exception is the polymerization reaction rate constant  $\tilde{r}_g$  in (3), for which we made a qualified guess. This variable does not influence any steady-state solution and for the dynamic behaviour presented in Sec. 5, we investigate a wide range of values and can conclude that the nominal value seems to be reasonable.

**Table 1** Parameter values.

Parameter	Nominal value	Interval	Unit
$a$	1	0.5–3.0	$10^{-8}$ m/s
$D$	10	1–25	$10^{-12}$ m <sup>2</sup> /s
$g$	5	2.5–40	$10^{-7}$ s <sup>-1</sup>
$l_c$	4	1–1000	$10^{-6}$ m
$s_g$	2.121	0.5–4.0	$10^{-7}$ m/s
$r_g$	1.783	0.9–7.2	$10^{-5}$ m <sup>4</sup> /(mol s)
$\tilde{r}_g$	0.053	0.015–0.240	s <sup>-1</sup>
$c_c^\infty$	11.90	2.80–23.57	$10^{-3}$ mol/m <sup>3</sup>
$c_s(t)$	—	5.95–23.80	$10^{-3}$ mol/m <sup>3</sup>

### 3 Model transformation

As a first step in the construction of an efficient numerical method we scale the model (1) both in space and time. The former scaling allows us to work in a constant spatial domain in contrast to the varying domain defined by (1). The latter scaling grants a time adaptivity which is well needed due to the huge differences in axon growth rates occurring during simulations, cf. Sec. 5.2.

#### 3.1 Scaling in space

As the axon grows (or shrinks) the domain of the PDE (1a) expands (or contracts). Thus, straightforward application of an off-the-shelf numerical method is not possible. This issue is considered by McLean and Graham (2004) who made a spatial scaling transforming the domain of the PDE into the constant interval  $(0, 1)$ . As a consequence the same number of spatial computational cells can be used along the axon regardless of its length. Compare also with Diehl et al (2014); Graham et al (2006), where numerical computations are performed using this technique. The spatial scaling is the following:

$$y := \frac{x}{l(t)}, \quad \frac{\partial y}{\partial x} = \frac{1}{l(t)}, \quad \frac{\partial y}{\partial t} = -\frac{x l'(t)}{l(t)^2} = -\frac{y l'(t)}{l(t)},$$

where  $x \in [0, l(t)]$  and thus  $y \in [0, 1]$ . With  $\bar{c}(y, t) := c(y l(t), t)$ , the derivatives can be written as

$$\frac{\partial c}{\partial x} = \frac{1}{l(t)} \frac{\partial \bar{c}}{\partial y}, \quad \frac{\partial^2 c}{\partial x^2} = \frac{1}{l(t)^2} \frac{\partial^2 \bar{c}}{\partial y^2}, \quad \frac{\partial c}{\partial t} = \frac{\partial \bar{c}}{\partial t} - \frac{y l'(t)}{l(t)} \frac{\partial \bar{c}}{\partial y}.$$

Substituting these into the equations and noting that

$$\frac{a - y l'(t)}{l(t)} = \frac{a - y r_g (c_c(t) - c_c^\infty)}{l(t)},$$

we can write the transformed dynamic model (1) as

$$\left\{ \begin{aligned} \frac{\partial \bar{c}}{\partial t} + (a - y r_g (c_c - c_c^\infty)) \frac{1}{l} \frac{\partial \bar{c}}{\partial y} - D \frac{1}{l^2} \frac{\partial^2 \bar{c}}{\partial y^2} &= -g \bar{c}, \end{aligned} \right. \quad (8a)$$

$$\left\{ \begin{aligned} \frac{dc_c}{dt} &= \frac{(a - g l_c)}{l_c} c_c - \frac{D}{l_c} \frac{1}{l} \bar{c}_y \\ &\quad - \frac{(r_g c_c + \tilde{r}_g l_c)(c_c - c_c^\infty)}{l_c}, \end{aligned} \right. \quad (8b)$$

$$\left\{ \begin{aligned} \frac{dl}{dt} &= r_g (c_c - c_c^\infty), \end{aligned} \right. \quad (8c)$$

$$\bar{c}(0, t) = c_s(t), \quad (8d)$$

$$\bar{c}(1, t) = c_c(t), \quad (8e)$$

$$\bar{c}(y, 0) = c^0(y l^0), \quad (8f)$$

$$c_c(0) = c^0(l^0), \quad (8g)$$

$$l(0) = l^0, \quad (8h)$$

for  $y \in (0, 1)$  and  $t > 0$ .

#### 3.2 Scaling in time and space

For short axon lengths the advection and diffusion effects are large in relation to the domain. This is reflected by the coefficients  $1/l$  and  $1/l^2$  in (8a). Thus the model can be expected to be substantially more difficult to simulate when the axon length is small. Additionally, recall that the axon length is expected to change multiple orders of magnitude

during its growth. Therefore we would expect that considerably longer time steps can be taken when  $l$  is large compared to the short steps needed to resolve the fast evolution when  $l$  is small. Our temporal scaling is designed to implement such a desired time adaptivity.

First, we make the assumption that  $l(t) > 0$  for all  $t \geq 0$ , define the dimensionless function

$$\Gamma(t) := a \int_0^t \frac{ds}{l(s)} \quad (9)$$

and introduce the following coordinate transformation, to be applied on (1):

$$\begin{cases} y := \frac{x}{l(t)}, \\ \tau := \Gamma(t), \end{cases} \quad 0 \leq x \leq l(t), t \geq 0. \quad (10)$$

Since  $\Gamma'(t) = a/l(t) > 0$  for all  $t \geq 0$ , the inverse of  $\Gamma$  exists, and (10) is equivalent to

$$\begin{cases} x = y\bar{l}(\tau), \\ t = \Gamma^{-1}(\tau), \end{cases} \quad 0 \leq y \leq 1, \tau \geq 0, \quad (11)$$

where  $\bar{l}(\tau) := l(\Gamma^{-1}(\tau)) = l(t)$ . It is convenient to introduce the notation  $\bar{t}$  for  $\Gamma^{-1}$ , i.e.  $t = \bar{t}(\tau) := \Gamma^{-1}(\tau)$ . Furthermore, differentiating the identity  $\Gamma(\bar{t}(\tau)) = \tau$  gives  $\Gamma'(\bar{t}(\tau))\bar{t}'(\tau) = 1$  which results in an ODE to update the original time:

$$\frac{d\bar{t}}{d\tau} = \frac{1}{\Gamma'(\bar{t}(\tau))} = \frac{1}{a/l(\bar{t}(\tau))} = \frac{\bar{l}(\tau)}{a}.$$

We append this ODE to the dynamical system (1). Furthermore, we set  $\bar{c}_c(\tau) := c_c(t)$ , and redefine  $\bar{c}(y, \tau) := c(x, t)$ . We have

$$\begin{cases} \frac{\partial \tau}{\partial t} = \Gamma'(t) = \frac{a}{l(t)} = \frac{a}{\bar{l}(\tau)}, & \frac{\partial \tau}{\partial x} = 0, \\ c'_c(t) = \bar{c}'_c(\tau) \frac{\partial \tau}{\partial t} = \frac{a \bar{c}'_c(\tau)}{\bar{l}(\tau)}, & l'(t) = \frac{a \bar{l}'(\tau)}{\bar{l}(\tau)}, \\ \frac{\partial y}{\partial t} = -\frac{x l'(t)}{l(t)^2} = -\frac{a y \bar{l}'(\tau)}{\bar{l}(\tau)^2}, & \frac{\partial y}{\partial x} = \frac{1}{l(t)} = \frac{1}{\bar{l}(\tau)}. \end{cases}$$

Furthermore,

$$\begin{cases} \frac{\partial c}{\partial x} = \frac{1}{\bar{l}(\tau)} \frac{\partial \bar{c}}{\partial y}, & \frac{\partial^2 c}{\partial x^2} = \frac{1}{\bar{l}(\tau)^2} \frac{\partial^2 \bar{c}}{\partial y^2}, \\ \frac{\partial c}{\partial t} = \frac{a}{\bar{l}(\tau)} \frac{\partial \bar{c}}{\partial \tau} - \frac{a y \bar{l}'(\tau)}{\bar{l}(\tau)^2} \frac{\partial \bar{c}}{\partial y}. \end{cases}$$

To simplify notation we note that

$$1 - y \frac{\bar{l}'}{\bar{l}} = 1 - y \frac{l'}{a} = 1 - y \frac{r_g}{a} (\bar{c}_c - c_c^\infty)$$

and define the functions

$$\alpha(\bar{c}_c, y) := 1 - y \frac{r_g}{a} (\bar{c}_c - c_c^\infty) \quad \text{and}$$

$$\beta(\bar{c}_c, \bar{l}) := \frac{(a - g l_c) \bar{l} \bar{c}_c - \bar{l} (r_g \bar{c}_c + \tilde{r}_g l_c) (\bar{c}_c - c_c^\infty)}{a l_c}.$$

Thus, after both space and time scaling of (1) we get the dynamic system

$$\begin{cases} \frac{\partial \bar{c}}{\partial \tau} + \alpha(\bar{c}_c, y) \frac{\partial \bar{c}}{\partial y} - \frac{D}{a} \frac{1}{\bar{l}} \frac{\partial^2 \bar{c}}{\partial y^2} = -\frac{g}{a} \bar{l} \bar{c}, \end{cases} \quad (12a)$$

$$\frac{d\bar{c}_c}{d\tau} = \beta(\bar{c}_c, \bar{l}) - \frac{D}{a l_c} \bar{c}_c^-, \quad (12b)$$

$$\frac{d\bar{l}}{d\tau} = \frac{r_g}{a} \bar{l} (\bar{c}_c - c_c^\infty), \quad (12c)$$

$$\frac{d\bar{r}}{d\tau} = \frac{1}{a} \bar{l}, \quad (12d)$$

$$\bar{c}(0, \tau) = c_s(\bar{r}(\tau)), \quad (12e)$$

$$\bar{c}(1, \tau) = \bar{c}_c(\tau), \quad (12f)$$

$$\bar{c}(y, 0) = c^0(y l^0), \quad (12g)$$

$$\bar{c}_c(0) = c^0(l^0), \quad (12h)$$

$$\bar{l}(0) = l^0, \quad (12i)$$

$$\bar{r}(0) = 0. \quad (12j)$$

which is defined for  $y \in (0, 1)$  and  $\tau > 0$ .

The positive effect of the time scaling for short axon lengths can for example be seen by comparing (8a) with (12a). In the latter there is one less reciprocal of  $l$  in the advection and diffusion coefficients. Thus, the evolution of the system when the axon length  $l$  is small is easier to resolve in the scaled time  $\tau$  compared to the original time  $t$ .

Note that the PDE (12a) is linear in  $\bar{c}$ . However, the coefficients in (12a) and the boundary conditions (12e)–(12f) depend on  $\bar{c}_c$ ,  $\bar{l}$ , and  $\bar{r}$  which are determined by the nonlinear ODEs (12b)–(12d). Since, additionally the ODE (12b) depends on  $\bar{c}_y^-$ , the model (12) defines a fully coupled nonlinear system. The same observations can be made for the system (8). In Sec. 4 we will apply a numerical method that decouples the approximations of the differential equations such that the aforementioned linearities can be utilized.

Finally, we comment on an alternative time scaling where (9) is replaced by

$$\Gamma_D(t) := D \int_0^t \frac{ds}{l(s)^2}, \quad (13)$$

which means that the former scaling with respect to advection velocity  $a$  is replaced by a scaling with respect to the diffusion  $D$ . This gives instead of (12a) the PDE

$$\frac{\partial \bar{c}}{\partial \tau} + \frac{a}{D} \alpha(\bar{c}_c, y) \bar{l} \frac{\partial \bar{c}}{\partial y} - \frac{\partial^2 \bar{c}}{\partial y^2} = -\frac{g}{D} \bar{l}^2 \bar{c}. \quad (14)$$

The corresponding ODEs are similarly given by multiplying the right-hand sides of (12b)–(12d) by  $a\bar{l}/D$ . The function  $\Gamma_D$  implements a more aggressive scaling promoting high resolution when the axon is short at the expense of poorer resolution during the time periods when  $l$  is large. See also the remarks at the end of Sec. 5.4.

#### 4 Numerical methods

We approximate the fully scaled system (12) using the method of lines (MOL). To this end, we perform a spatial discretization in Sec. 4.1, which is followed by temporal discretizations in Sec. 4.2. Note that the same discretizations may be applied to system (8) which is only scaled in space. However, in Sec. 5.4 we will see that by using time scaling we largely gain in efficiency and reliability. See also Diehl et al (2014) for an explicit Euler discretization of (8).

##### 4.1 Spatial discretization

The spatial interval  $[0, 1]$  is divided into  $M$  subintervals of size  $\Delta y := 1/M$ . The grid points are located at  $y_j := j\Delta y$ ,  $j = 0, \dots, M$ . In particular,  $y_M = 1$  holds. To each grid point we associate a concentration value  $\tilde{c}_j = \tilde{c}_j(\tau) \approx \bar{c}(y_j, \tau)$ . We approximate the spatial derivatives of  $\bar{c}$  in the PDE (12a) by second-order central finite differences, i.e., for  $j = 1, \dots, M-1$ ,

$$\begin{aligned} \frac{\partial \bar{c}}{\partial y}(y_j, \cdot) &\approx \frac{\tilde{c}_{j+1} - \tilde{c}_{j-1}}{2\Delta y} \quad \text{and} \\ \frac{\partial^2 \bar{c}}{\partial y^2}(y_j, \cdot) &\approx \frac{\tilde{c}_{j+1} - 2\tilde{c}_j + \tilde{c}_{j-1}}{(\Delta y)^2}. \end{aligned} \quad (15)$$

Thus, the PDE (12a) is transformed into a system of  $M-1$  MOL ODEs:

$$\begin{aligned} \frac{d\tilde{c}_j}{d\tau} &= -\alpha(\tilde{c}_c, y_j) \frac{\tilde{c}_{j+1} - \tilde{c}_{j-1}}{2\Delta y} \\ &+ \frac{D}{a} \frac{1}{\bar{l}} \frac{\tilde{c}_{j+1} - 2\tilde{c}_j + \tilde{c}_{j-1}}{(\Delta y)^2} - \frac{g}{a} \tilde{l} \tilde{c}_j, \quad j = 1, \dots, M-1. \end{aligned}$$

Here  $\tilde{c}_0(\tau)$  and  $\tilde{c}_M(\tau)$  should be interpreted as  $c_s(\bar{l}(\tau))$  and  $\bar{c}_c(\tau)$ , respectively, according to the boundary conditions (12e)–(12f). In the cone concentration ODE (12b) we use a one-sided second-order approximation together with the continuity boundary condition (12f) to get:

$$\bar{c}_c^- \approx \frac{3\tilde{c}_M - 4\tilde{c}_{M-1} + \tilde{c}_{M-2}}{2\Delta y} = \frac{3\tilde{c}_c - 4\tilde{c}_{M-1} + \tilde{c}_{M-2}}{2\Delta y}. \quad (16)$$

The above efforts combined replace the model (12) by a system of  $M+2$  ODEs. This MOL discretization can be written

on matrix form as

$$\begin{cases} \frac{d\tilde{\mathbf{c}}}{d\tau} = \mathbf{A}(\tilde{\mathbf{u}}, \mathbf{y})\tilde{\mathbf{c}} + \mathbf{B}(\tilde{\mathbf{u}}), \\ \frac{d\tilde{\mathbf{u}}}{d\tau} = \mathbf{F}(\tilde{\mathbf{c}}, \tilde{\mathbf{u}}), \end{cases} \quad (17a)$$

$$(17b)$$

with initial values given by (12g)–(12j) and a sampling of  $c^0$ :  $\tilde{c}_j(0) = c^0(y_j l^0)$  for  $j = 1, \dots, M-1$ . We explain the notation of (17) in what follows. To this end, introduce the vectors

$$\mathbf{y} := (y_1 \ y_2 \ \dots \ y_{M-1})^T,$$

$$\tilde{\mathbf{c}}(\tau) := (\tilde{c}_1(\tau) \ \tilde{c}_2(\tau) \ \dots \ \tilde{c}_{M-1}(\tau))^T,$$

$$\tilde{\mathbf{u}}(\tau) := (\tilde{c}_c(\tau) \ \bar{l}(\tau) \ \bar{i}(\tau))^T.$$

and define the tridiagonal matrix

$$\mathbf{A} := \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & a_{M-2,M-3} & a_{M-2,M-2} & a_{M-2,M-1} \\ 0 & \dots & 0 & a_{M-1,M-2} & a_{M-1,M-1} \end{pmatrix}$$

of size  $(M-1) \times (M-1)$ . The entries of  $\mathbf{A} = \mathbf{A}(\tilde{\mathbf{u}}, \mathbf{y})$  depend on the solutions of the ODEs (12b)–(12d). These entries are given by

$$\begin{aligned} a_{j,j-1}(\tilde{\mathbf{u}}, \mathbf{y}) &:= \frac{\alpha(\tilde{c}_c, y_j)}{2\Delta y} + \frac{D}{a(\Delta y)^2} \frac{1}{\bar{l}}, \quad j = 2, \dots, M-1, \\ a_{j,j}(\tilde{\mathbf{u}}, \mathbf{y}) &:= -\frac{2D}{a(\Delta y)^2} \frac{1}{\bar{l}} - \frac{g}{a} \bar{l}, \quad j = 1, \dots, M-1, \\ a_{j,j+1}(\tilde{\mathbf{u}}, \mathbf{y}) &:= -\frac{\alpha(\tilde{c}_c, y_j)}{2\Delta y} + \frac{D}{a(\Delta y)^2} \frac{1}{\bar{l}}, \quad j = 1, \dots, M-2. \end{aligned}$$

Note how the sub and super diagonals vary with  $y_j$ : on row  $j$  we input  $y_j$  in  $\alpha$  to get the correct matrix elements. Also note that on the first and last row there are only two non-zero elements. Further, define the solution-dependent vector

$$\mathbf{B}(\tilde{\mathbf{u}}) := \begin{pmatrix} \left( \frac{1}{2\Delta y} \alpha(\tilde{c}_c, y_1) + \frac{D}{a(\Delta y)^2} \frac{1}{\bar{l}} \right) c_s(\bar{l}) \\ 0 \\ \vdots \\ 0 \\ \left( -\frac{1}{2\Delta y} \alpha(\tilde{c}_c, y_{M-1}) + \frac{D}{a(\Delta y)^2} \frac{1}{\bar{l}} \right) \bar{c}_c \end{pmatrix},$$

which is of size  $M-1$  and contains the boundary conditions (12e)–(12f). Finally, define the vector

$$\mathbf{F}(\tilde{\mathbf{c}}, \tilde{\mathbf{u}}) := \begin{pmatrix} \beta(\tilde{c}_c, \bar{l}) - \frac{D}{a\bar{l}_c} \frac{3\tilde{c}_c - 4\tilde{c}_{M-1} + \tilde{c}_{M-2}}{2\Delta y} \\ \frac{r_g}{a} \bar{l}(\tilde{c}_c - c_c^\infty) \\ \frac{1}{a} \bar{l} \end{pmatrix}$$

corresponding to the right-hand sides of the ODEs (12b)–(12d). Thus, we arrive at the MOL discretization (17), given by applying second-order finite differences to the system (12).

## 4.2 Full discretizations

Based on the semi-discretization (17) we use the explicit Euler and Peaceman-Rachford methods to construct two different full discretizations. Denote the time step by  $\Delta\tau$  and let  $\tau^n := n\Delta\tau$ ,  $n = 0, 1, \dots, N$ , where  $N$  is the number of time steps used. At time  $\tau = \tau^n$ , the concentration within the axon is approximated by the numerically computed values  $C_j^n \approx \tilde{c}_j(\tau^n) \approx \bar{c}(y_j, \tau^n)$ ,  $j = 1, \dots, M-1$ . The approximate growth-cone concentration is denoted by  $C_c^n \approx \bar{c}_c(\tau^n)$ , the approximate axon length by  $L^n \approx \bar{l}(\tau^n)$ , and the approximate (original) time by  $t^n \approx \bar{t}(\tau^n)$ . We gather these values in the vectors

$$\mathbf{C}^n := (C_1^n \ C_2^n \ \dots \ C_{M-1}^n)^T,$$

$$\mathbf{U}^n := (C_c^n \ L^n \ t^n)^T.$$

By applying the explicit Euler temporal discretization to the spatial semi-discretization (17) we get, for  $n = 0, 1, \dots, N-1$ ,

$$\mathbf{C}^{n+1} = \mathbf{C}^n + \Delta\tau (\mathbf{A}(\mathbf{U}^n, \mathbf{y})\mathbf{C}^n + \mathbf{B}(\mathbf{U}^n)), \quad (18a)$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta\tau \mathbf{F}(\mathbf{C}^n, \mathbf{U}^n). \quad (18b)$$

Equation (18) defines a time-marching scheme with initial values

$$C_j^0 := c^0(y_j l^0), \quad j = 1, \dots, M-1, \quad (19a)$$

$$\mathbf{U}^0 := (c^0(l^0) \ l^0 \ 0)^T. \quad (19b)$$

In the original coordinates, we have  $c_c(t^n) \approx C_c^n$  and  $l(t^n) \approx L^n$  at the time points  $t^n$  and for the concentration distribution in the axon  $c(x_j^n, t^n) \approx C_j^n$  where  $x_j^n := y_j L^n = j\Delta y L^n$ .

The explicit Euler method (18) defines computations that are simple to implement since only old values of the unknowns are used. However, a problem with this method (and explicit methods in general) is that, given  $\Delta y = 1/M$ , the time step  $\Delta\tau$  has to be chosen small to avoid numerical instabilities. When diffusion is present these time step restrictions are very prohibitive and result in large CPU times. In the classical analysis of the explicit Euler method applied to diffusion–advection–reaction equations a so called CFL condition must be fulfilled to have stability. Here we have an additional complication due to the coupling to the ODEs (12b)–(12d). Assume that the scheme (18) produces a numerical solution that satisfies

$$L^n \geq l^{\min} > 0 \quad \text{for } n = 0, 1, \dots, N, \quad (20)$$

where  $l^{\min}$  is a constant. Then the explicit Euler method (18) has the CFL stability criterion

$$\Delta\tau \leq \frac{a}{2D} l^{\min} (\Delta y)^2. \quad (21)$$

To summarize, while the advantage of the explicit scheme (18) is its simple implementation, the disadvantage is the long computation times required due to the factor  $(\Delta y)^2$  in the right-hand side of (21). With  $l^{\min} = 1\mu\text{m}$  and the parameter values of Table 1, the CFL condition (21) is for  $\Delta y = 1/M = 1/100$  given by  $\Delta\tau \leq 5 \cdot 10^{-8}$ . This should be compared with the long simulation times usually needed to get close to steady state. See for example the simulation presented in Fig. 2 with end time  $t^N \geq 6 \cdot 10^8 \text{ s} \approx 19$  years and where the axon length is at its longest  $L^n \approx 0.08 \text{ m}$ . Using (12d) we can deduce that  $\Delta t$  is, at its largest, approximately 0.4 s. This means that we need more than  $15 \cdot 10^8$  time steps just to ensure the stability of explicit Euler. Additionally, if we want higher accuracy in space (as in the aforementioned simulation) the number of time steps  $N$  needed grows quadratically with  $M$  at the same time as there are  $M$  function computations at each time point. In other words, with explicit Euler time stepping, halving  $\Delta y$ , means that the CPU time increases with a factor approximately  $2^3 = 8$ .

To avoid the aforementioned problems we can instead use an unconditionally stable implicit scheme, like the implicit Euler method. Such a method needs no stability restriction on  $\Delta\tau$ . However, for an implicit method the semi-discretization (17a) defines a large system of equations which is coupled with the nonlinear ODEs of (17b). This means that in every time step a nonlinear equation solver needs to be applied to the full system.

We propose to instead use the Peaceman-Rachford splitting method for time discretization, cf. Douglas (1955); Peaceman and Rachford (1955); Hundsdorfer and Verwer (2003); Hansen and Henningsson (2013). This method needs neither a stability constraint on  $\Delta\tau$  (as a function of  $\Delta y$ ), nor the numerical solution of a large nonlinear system at each time step. Furthermore, while the explicit and implicit Euler methods are first-order accurate, the Peaceman-Rachford method is second-order accurate. Finally, in contrast to many other splitting methods, the Peaceman–Rachford scheme preserves the steady states of the system that it approximates, cf. Hundsdorfer and Verwer (2003, Sec. IV.3.1). This preservation property is of utmost importance for our investigations in Sec. 5.5.

Taking one time step of size  $\Delta\tau$  using the Peaceman-Rachford splitting scheme consists of, in sequence, solving for  $\mathbf{U}^{n+1/2}$ ,  $\mathbf{C}^{n+1/2}$ ,  $\mathbf{C}^{n+1}$ , and  $\mathbf{U}^{n+1}$ , respectively, in the following equations:

$$\mathbf{U}^{n+1/2} = \mathbf{U}^n + \frac{\Delta\tau}{2} \mathbf{F}(\mathbf{C}^n, \mathbf{U}^n), \quad (22a)$$

$$\mathbf{C}^{n+1/2} = \mathbf{C}^n + \frac{\Delta\tau}{2} (\mathbf{A}(\mathbf{U}^{n+1/2}, \mathbf{y})\mathbf{C}^{n+1/2} + \mathbf{B}(\mathbf{U}^{n+1/2})), \quad (22b)$$

$$\mathbf{C}^{n+1} = \mathbf{C}^{n+1/2} + \frac{\Delta\tau}{2} (\mathbf{A}(\mathbf{U}^{n+1/2}, \mathbf{y})\mathbf{C}^{n+1/2} + \mathbf{B}(\mathbf{U}^{n+1/2})), \quad (22c)$$

$$\mathbf{U}^{n+1} = \mathbf{U}^{n+1/2} + \frac{\Delta\tau}{2} \mathbf{F}(\mathbf{C}^{n+1}, \mathbf{U}^{n+1}). \quad (22d)$$

Here,  $\mathbf{C}^n$  and  $\mathbf{U}^n$  are known from the previous time step or, for  $n = 0$ , from the initial conditions (19).

Some comments about the Peaceman–Rachford method are appropriate. First note that the updates (22a) and (22c) are explicit Euler steps similar to (18a) and (18b) and they are therefore cheap to compute. The other two updates, (22b) and (22d), are implicit Euler steps. However, they are implicit only in  $\mathbf{C}^{n+1/2}$  and  $\mathbf{U}^{n+1}$ , respectively. As a consequence, performing the update (22b) only amounts to solving the linear system of equations

$$\left( \mathbf{I} - \frac{\Delta\tau}{2} \mathbf{A}(\mathbf{U}^{n+1/2}, \mathbf{y}) \right) \mathbf{C}^{n+1/2} = \mathbf{C}^n + \frac{\Delta\tau}{2} \mathbf{B}(\mathbf{U}^{n+1/2}) \quad (23)$$

for  $\mathbf{C}^{n+1/2}$ , where  $\mathbf{I}$  is the  $(M-1) \times (M-1)$  identity matrix. This should be compared with a fully implicit method, for which the corresponding system to be solved in each time step is nonlinear (and also slightly larger, consisting of  $M+2$  equations). Finally, performing the update (22d) only means solving a system of three nonlinear equations. For that, a standard nonlinear equation solver, like Newton’s method, can be applied for rapid solution. Such a solver requires a tolerance parameter to determine how many local iterations are needed. However, since the nonlinear equation system (22d) is tiny compared to the linear system (22b) the solution of the former can be done in a negligible time. Thus, optimizing the tolerance parameter is superfluous; we may choose it sharp without significantly affecting the overall efficiency of the scheme.

For the update (22b) to be well-defined the matrix  $\mathbf{I} - \Delta\tau/2 \cdot \mathbf{A}(\mathbf{U}^{n+1/2}, \mathbf{y})$  must be invertible, we give a sufficient condition in the following lemma.

**Lemma 1** *Assume that the scheme (22) produces a numerical solution that satisfies  $L^{n+1/2} > 0$  and*

$$|C_c^{n+1/2} - c_c^\infty| \leq \gamma, \quad \text{for } n = 0, 1, \dots, N, \quad (24)$$

*and for some positive constant  $\gamma$ . Then, if*

$$\Delta\tau < \frac{4a}{r_g} \cdot \frac{1}{\gamma}, \quad (25)$$

*the linear system of equations (23) has a unique solution at each time step  $n$ .*

See Appendix A for a proof. Note that the condition (25) is well behaved in the sense that it is not affected by small values of  $\Delta y$ . In fact, consider the simulations performed in Sec. 5.2, the solution fulfils the bound (24) with  $\gamma = 11.9 \cdot 10^{-3} \text{ mol/m}^3$ . With this constant and with the parameter values as in Table 1 the time step condition (25) reads  $\Delta\tau < 0.18$ . We emphasize that using this bound on  $\Delta\tau$  does not guarantee that the entire numerical scheme works, e.g. the implicit Euler step (22d) for the ODEs may require smaller values on  $\Delta\tau$ . However, further investigations, cf.

Fig. 4, yield that much smaller time steps are needed for accuracy and mean no severe restriction in CPU time.

Note that if the scaling with diffusion  $D$  (13) is used instead of the one with advection  $a$  (9), the lemma should be changed in the following way: We have to assume that there is a constant  $l^{\max}$  such that  $L^{n+1/2} \leq l^{\max}$  for all  $n$ , and then the condition (25) is replaced by  $\Delta\tau < 4D/(r_g l^{\max} \gamma)$ .

## 5 Simulations

In this section we present numerical simulations performed with the methods in Sec. 4. The purpose is twofold: Firstly, in Secs 5.2–5.4 we examine the dynamics of the system (1) and how it affects the choice of numerical method. Secondly, in Sec. 5.5, we use the efficient Peaceman–Rachford discretization to perform parameters studies. That is, we vary the parameter values of Table 1 to investigate the sensitivity of the dynamical solution with respect to each parameter.

All results plotted in this section are numerical approximations, however, for the sake of brevity we will not use the notation of Sec. 4 but rather refer to each approximation via the continuous variable that is approximated. For example, in Fig. 2(a) a Peaceman–Rachford approximation  $L^n$ ,  $n = 0, 1, \dots, N$  of  $l$  given by (22) is plotted but the approximation is referred to by  $l$ .

The numerical schemes presented in Sec. 4 and used in the current section have been implemented in MATLAB (R2014a). The code is available from the ModelDB database with accession number 187687<sup>1</sup>.

### 5.1 Biological, physical, and numerical constants

In our investigations, biological and physical as well as numerical parameters will be varied depending on the inquiry at hand. The nominal parameter values and initial and boundary conditions given here and in Table 1 are used except when something else is explicitly stated. Further, when nothing else is stated, we use the Peaceman–Rachford discretization (22) to approximate the spatial semi-discretization (17) of the fully scaled system (12). Among the methods presented in this article this is by far the most efficient and accurate approximation of system (1), as we shall see in Sec. 5.3 and Sec. 5.4.

In all performed simulations the time-dependent soma concentration  $c_s(t)$  is chosen piecewise constant as

$$c_s(t) := \begin{cases} 2c_c^\infty = 23.80 \text{ mmol/m}^3, & 0 \leq t < 2 \cdot 10^8 \text{ s}, \\ \frac{c_c^\infty}{2} = 5.95 \text{ mmol/m}^3, & 2 \cdot 10^8 \text{ s} \leq t < 4 \cdot 10^8 \text{ s}, \\ 2c_c^\infty = 23.80 \text{ mmol/m}^3, & 4 \cdot 10^8 \text{ s} \leq t. \end{cases}$$

<sup>1</sup> URL: <https://senselab.med.yale.edu/ModelDB/showModel.cshtml?model=187687>



(26)

With this choice of  $c_s$  we will observe both axon expansion and contraction. Additionally, note that we initially have  $c_s(t)/c_c^\infty = 2$  which means that the axon will grow regardless of the initial length  $l^0$ , cf. Diehl et al (2014, Thm 4.1 and Fig. 9).

Since we are interested in the growth of the axon from a small length to its steady state (and possible contraction due to a decrease in the soma concentration), we choose a small initial length  $l^0 = 1 \mu\text{m}$ . For such small axon lengths it seems reasonable that the initial tubulin concentration  $c^0$  along the axon is constant and equal to the initial soma concentration. Therefore we make the simple choice  $c^0 = c^0(x) = 23.80 \cdot 10^{-3} \text{ mol/m}^3$  for the initial concentration profile.

In each of our investigations we specify an end time  $T$ . The simulations will be performed until the first  $n$  such that  $t^n \geq T$  and we denote this value of  $n$  by  $N$ . Due to the adaptivity in time caused by the time scaling, which depends on the solution, we cannot expect that  $t^N$  is equal to  $T$ . However approximations of  $c(x, T)$ ,  $c_c(T)$ , and  $l(T)$  can be found by simple interpolations.

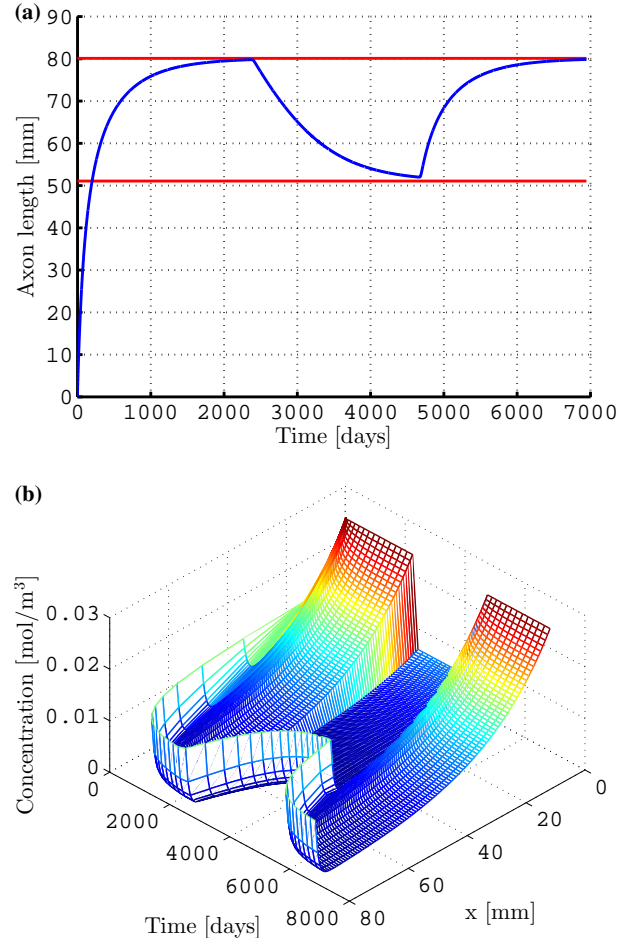
## 5.2 Dynamical properties of the model

In Sec. 3 we stated that system (1) exhibits dynamical phenomena on different time scales. In this section we verify this claim by showing numerical simulations on very fine grids as to minimize the influence of numerical artefacts on the exact dynamics of the systems. The numerical method, the biological parameters, and the initial data are chosen as described in Sec. 5.1. For the spatial discretization we use  $M = 10^4$  meaning  $\Delta y = 10^{-4}$ . (Recall that there is no CFL condition for the Peaceman–Rachford scheme).

To investigate the behaviour on large time scales and the convergence to steady state we choose the end time  $T = 6 \cdot 10^8 \text{ s} \approx 19 \text{ years}$  and use the small time step  $\Delta \tau = 5 \cdot 10^{-4}$ . The results are plotted in Fig. 2 where we can observe how the variations in the soma concentration  $c_s$  cause axon expansion as well as contraction.

In Fig. 3 we can observe the fast transient behaviour of system (1) for small times. Here the end time is chosen as  $T = 3600 \text{ s} = 1 \text{ h}$ , tiny compared to the previous simulation. Similarly, we here use a much smaller time step  $\Delta \tau = 10^{-5}$  than in Fig. 2. Also note that this choice of  $T$  gives constant  $c_s(t) = 2c_c^\infty$ . It is remarkable how close the cone concentration  $c_c$  is to its steady-state value  $c_c^\infty = 11.90 \cdot 10^{-3} \text{ mol/m}^3$  already after 1 h, whereas it takes more than a thousand days for the axon length  $l$  to come close to its steady state 80.10 mm.

The evolution on these different time scales is one of the major problems for a numerical method to manage and it is an important motivation for the introduction of the time



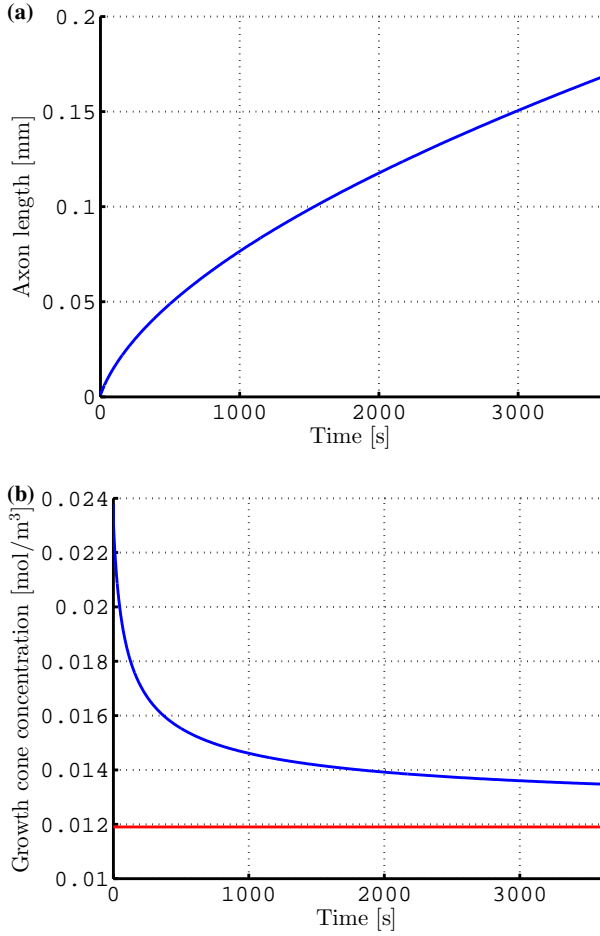
**Fig. 2** Axon growth and tubulin contraction as a result of a piecewise constant soma concentration  $c_s(t)$ , which can be seen in plot (b) at  $x = 0$ . In (a) the axon length  $l(t)$  is plotted in blue and the red lines are the steady-state lengths corresponding to the two different values of soma concentration. In (b) the tubulin concentration  $c(x, t)$  in the axon is plotted over time and space. Note the characteristic profile of the axon concentration, cf. Diehl et al (2014, Sec. 4). Further, note how slowly the axon length responds to changes in  $c_s(t)$  and how long time it takes for the axon to grow to its steady state. See also Figs. 13 and 15 in Appendix B for two-dimensional slices of (b) at different values of  $x$  and  $t$ .

scaling (10). Even with time scaling, the fast transient of  $c_c$  is the most difficult phenomenon for our numerical methods to resolve and therefore it is the main source of errors.

## 5.3 Accuracy of the scaled Peaceman–Rachford scheme

In this section we use numerical simulations to investigate the efficiency of the time discretization (22) as an approximation of model (1). We shine some light on why this approximation is more efficient than standard methods such as explicit Euler.

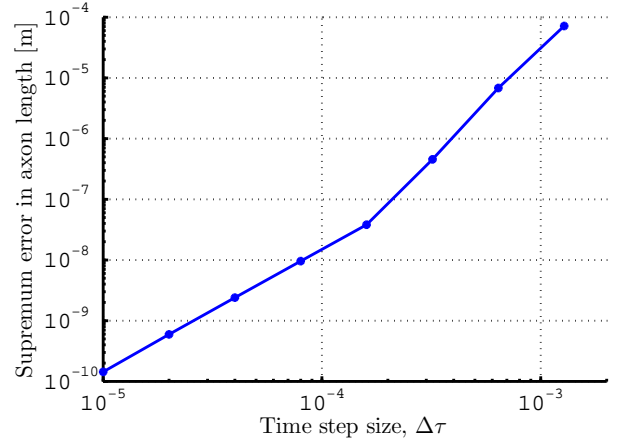
We first consider the end time  $T = 86400 \text{ s} = 1 \text{ day}$  so that we capture the transient of  $c_c$ . Note that this  $T$  means



**Fig. 3** (a) Axon length  $l(t)$  and (b) cone concentration  $c_c(t)$  during the first hour. In (a) we observe a fast growth of the axon to more than a hundred times its initial length. In (b) we see the fast convergence of the cone concentration  $c_c(t)$  from  $c_c(0) = 2c_c^\infty = 23.80 \cdot 10^{-3} \text{ mol/m}^3$  to its steady state  $c_c^\infty = 11.90 \cdot 10^{-3} \text{ mol/m}^3$  (marked with a red line).

that  $c_s$  is constant and equal to  $2c_c^\infty = 23.80 \cdot 10^{-3} \text{ mol/m}^3$ . Further, since we want to emphasize the error due to temporal discretization we choose a fine spatial grid  $M = 10^4$ ,  $\Delta y = 10^{-4}$ . The errors are approximated by comparing the numerical solutions with a reference solution. The latter is given by using the same discretization scheme (22) on a very fine grid both in time,  $\Delta \tau = 10^{-6}$ , and in space,  $\Delta y = 10^{-5}$ .

In Fig. 4 we see the results of a temporal convergence study for the Peaceman–Rachford discretization (22). That is, we consider a range of different time step sizes,  $\Delta \tau = 2^k \cdot 10^{-5}$ ,  $k = 0, \dots, 7$ , and perform simulations for each of them. Then, for each value of  $\Delta \tau$  the error in axon length  $l$  (given by taking the supremum norm over time) is plotted in Fig. 4 over the corresponding  $\Delta \tau$ . Considering the slope of the curve when  $\Delta \tau \lesssim 10^{-4}$  we observe the expected second-order convergence. The steeper slope for larger values of  $\Delta \tau$  can be explained by the fact that the transient of  $c_c$  is not properly resolved for those big time steps.

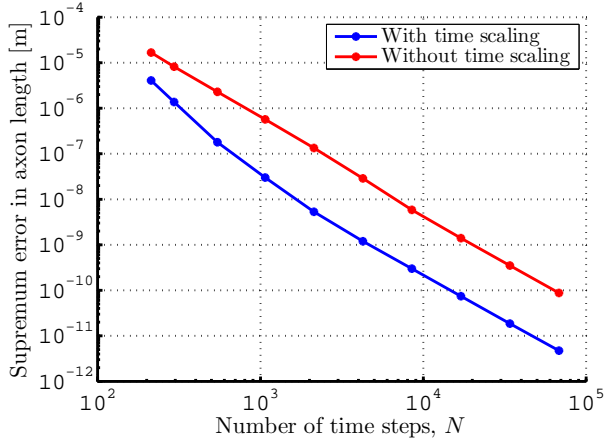


**Fig. 4** Temporal convergence plot for the Peaceman–Rachford scheme. Simulations are performed until the end time  $T = 86400 \text{ s} = 1 \text{ day}$ , during which the axon grows to a length of approximately 1.2 mm. For each value of  $\Delta \tau$ , the maximal error (over time) in the axon length is plotted. We observe the expected second-order convergence. For the leftmost point in the plot the CPU time is about five minutes and for the rightmost it is about three seconds (on a normal desktop computer).

#### 5.4 The need for time transformation

Recall the discussion in the beginning of Sec. 4, where we concluded that we can approximate the spatially scaled system (8) with finite differences and Peaceman–Rachford using the same procedure as when approximating the fully scaled system (12) in Sec. 4. This gives a numerical scheme without the adaptivity in time granted by the time scaling (9). When the simulation presented in Fig. 4 is repeated with this approximation no reasonable results are obtained due to the large time steps taken during the first parts of the simulations. This shows the strength of using the time scaling (9) to get a finer temporal resolution for small axon lengths.

When time scaling is not used, really small time steps are needed to properly resolve the transient of the cone concentration  $c_c$ . In Fig. 5 results are plotted from simulations similar to those presented in Fig. 4, but with a really small end time  $T = 60 \text{ s}$ . This  $T$  is chosen to be able to complete simulations with small enough time step sizes within reasonable CPU times. Also a finer reference solution is needed: we construct it in the same way as described above but here with  $\Delta \tau = 10^{-7}$ . To be able to compare the errors given when time scaling is used with those given when it is not we plot the errors over the number of time steps used, see Fig. 5. For each  $\Delta \tau = 2^k \cdot 10^{-6}$ ,  $k = 1, \dots, 10$ , we perform simulations using time scaling and for each value of  $\Delta \tau$  we store the number of time steps used  $N$ . Then, for each  $N$ , a simulation is performed without time scaling using  $N$  time steps. We recall that in model (8)  $t$  is the independent time variable, which gives us  $\Delta t = T/N$  for the latter simulations.

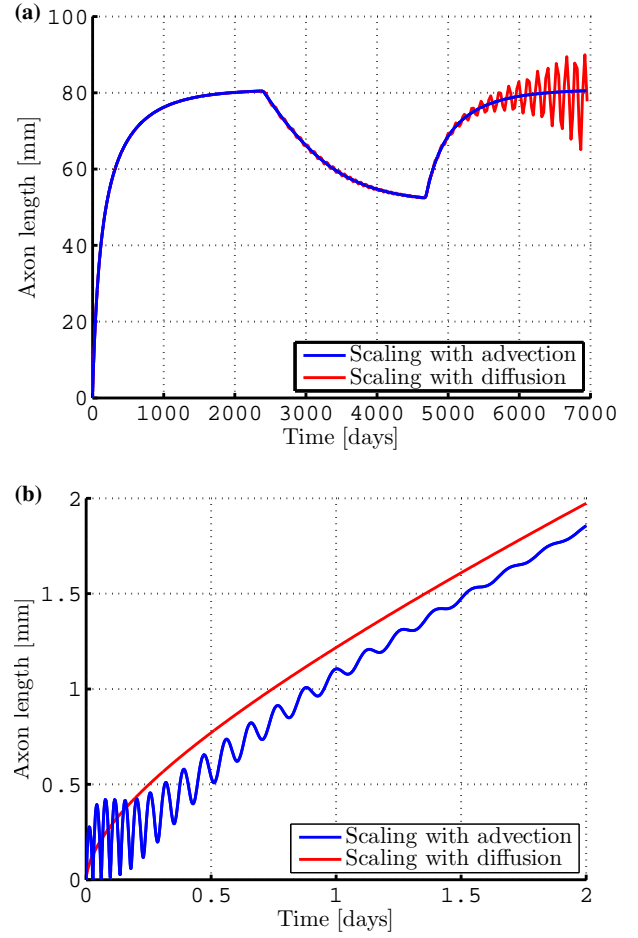


**Fig. 5** Temporal convergence plots for the Peaceman-Rachford scheme applied to the systems (8) (red) and (12) (blue), respectively. Simulations are performed with the small end time  $T = 60$  s giving a final axon length of approximately  $11 \mu\text{m}$ . For each discretization and each value of  $\Delta\tau$  the maximal error (over time) in axon length is plotted. With this fine resolution we can observe second-order convergence also when time scaling is not used.

The results in Fig. 5 show that we can achieve second-order convergence for the Peaceman-Rachford scheme even without using time scaling. However, the size of the time steps needed are too small for this discretization to be of any practical use. Therefore time scaling is preferable and this is what we use for our parameter studies in Sec. 5.5.

The simulations presented in Fig. 5 can be performed using the explicit Euler method (18). However, even when using time scaling, we would need to choose  $\Delta\tau < 5 \cdot 10^{-12}$  to fulfil the CFL condition (21). With our implementation running on an ordinary desktop computer, such a simulation would take about a quarter of a year to perform whereas the Peaceman-Rachford simulation with  $\Delta\tau = 1024 \cdot 10^{-6}$  takes half a second and that with  $\Delta\tau = 2 \cdot 10^{-6}$  takes about two minutes. Thus, explicit Euler is not an efficient choice, neither is implicit Euler due to its expensive time steps, cf. Sec. 4.2. The same conclusions may be drawn for any explicit method or implicit method that considers the whole axon growth model at once.

Finally, we give a short comment on the effects of using time scaling with  $D$  (13) instead of with  $a$  (9). We test both scalings using the settings described in Sec. 5.1 with  $\Delta y = 10^{-3}$  and  $T = 6 \cdot 10^8 \text{ s} \approx 19$  years. For the advection scaling we choose  $\Delta\tau = 6 \cdot 10^{-3}$  and for the diffusion scaling  $\Delta\tau = 4 \cdot 10^{-3}$ . These time steps are chosen large so that observable errors are produced. Furthermore, these choices mean that approximately the same number of time steps are used by both methods (approximately  $16 \cdot 10^3$  steps), i.e. about the same amount of CPU time is used by each (a couple of seconds). The results are plotted in Fig. 6 where we observe that the relative errors when using respective scaling are approximately the same. However, as one would ex-

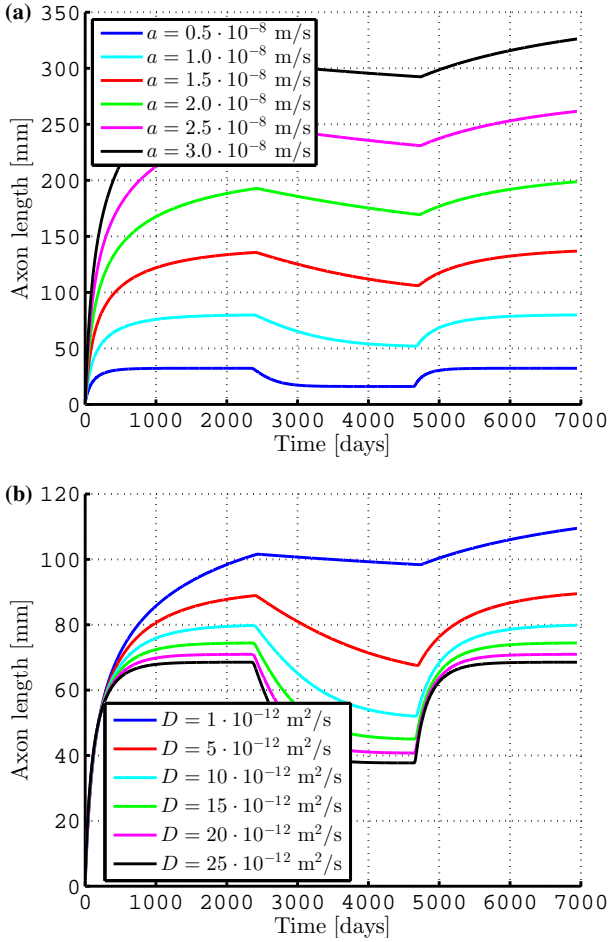


**Fig. 6** Plot (a) shows that both scalings yield similar large-scale growth, but the diffusion scaling may develop errors for large axons. Zooming in at short times and lengths, plot (b) reveals that errors may arise for the advection scaling.

pect the temporal location of the errors differ: The scaling with  $a$  has its largest (relative) errors when the axon is short, whereas the scaling with  $D$  has its largest errors when the axon is long. For the latter scaling we could also observe from our simulations that, after half the time steps were used, less than 9 s of simulated time  $t$  had passed and the axon had only grown from  $1 \mu\text{m}$  to  $2.7 \mu\text{m}$ . As we are interested in the convergence to steady state we will continue to use the scaling with advection (9) for our parameter studies in the upcoming section.

## 5.5 Parameter studies

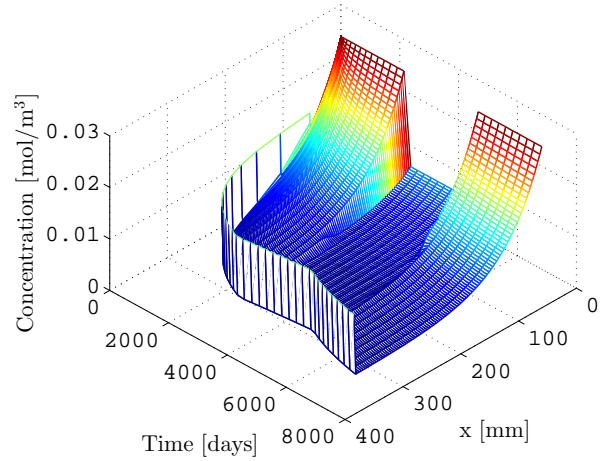
We shall now use the efficient and reliable Peaceman-Rachford scheme (22) to investigate the dependency of the axon growth dynamics with respect to the parameters of the model. That is, we vary one, or a few, parameters at a time performing simulations for a range of different values while keeping the other parameters constant at their nominal val-



**Fig. 7** Axon length  $l(t)$  for varying  $a$  and  $D$ .

ues. The parameter study in this section may be regarded as an extension of those performed in Diehl et al (2014, Sec. 5) for steady-state solutions, and the interested reader may benefit from having those results at hand. In that work, the steady states were presented for different parameter values, here we consider the dynamical convergence to these steady states. For easy comparison similar ranges of parameter values are used whenever applicable. We focus on the axon length  $l$  since the tubulin concentrations tend fast to their steady-state appearances: the concentration along the axon  $c$ , to its characteristic profile, cf. Fig. 2(b), and the concentration in the growth cone,  $c_c$ , to  $c_c^\infty$ , cf. Fig. 3(b).

Given the default parameter values in Table 1, each is varied and numerical simulations are performed with the settings specified in Sec. 5.1. The end time is chosen large,  $T = 6 \cdot 10^8 \text{ s} \approx 19$  years, since the convergence to steady state is of interest. The time step  $\Delta \tau = 5 \cdot 10^{-4}$  means both fast simulations and an initial temporal error of insignificant size, cf. Fig. 4. Also note that the system (1) by and large describes a parabolic problem. For such problems the initial errors typically have a small effect on the long time behaviour of the system. For the spatial resolution,  $\Delta y = 10^{-3}$



**Fig. 8** Tubulin concentration  $c(x,t)$  along the axon for  $a = 3.0 \cdot 10^{-8}$  m/s. See also Figs. 14 and 16 in Appendix B for two-dimensional slices at different values of  $x$  and  $t$ .

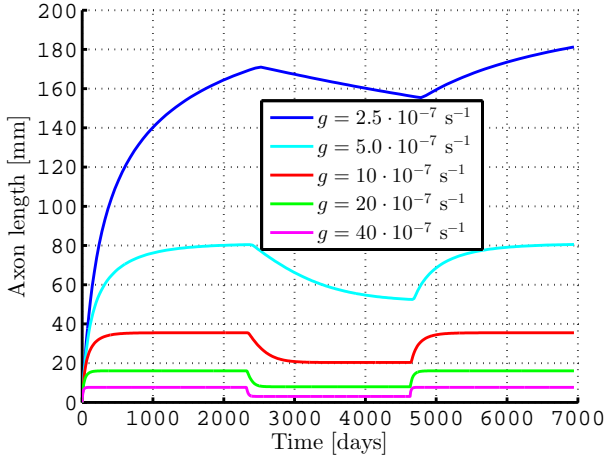
suffices for all simulations except for the extreme values of the parameters  $a$  and  $D$ .

In Fig. 7(a) the axon length  $l$  is plotted over time for varying values of the advection velocity  $a$ . We observe that the axon length is sensitive to variations in the active transport  $a$  with the length increasing with the transport velocity. However, the profile of the growth is rather insensitive to these variations. In Fig. 8 the tubulin concentration along the axon is plotted as a function of time and space for  $a = 3.0 \cdot 10^{-8}$  m/s. Observe the sharp gradient close to the growth cone, compare with Fig. 2(b) where  $a = 1.0 \cdot 10^{-8}$  m/s and with the concentration profiles in Diehl et al (2014, Fig. 7(a)).

In Fig. 7(b) we observe that  $l$  grows longer for small values of the diffusion coefficient  $D$  and that the axon length is not as sensitive to variations in  $D$  as it is to variations in  $a$ . However, the growth profile is more sensitive to changes in  $D$ : for large values of  $D$  the length reaches values close to its steady state faster than it does for smaller  $D$ . This can best be seen during and after the drop in soma concentration  $c_s$ . For small values of  $D$ , just as for big values of  $a$ , a sharp gradient in  $c$  is quickly created close to the growth cone, cf. Diehl et al (2014, Fig. 7). To resolve these gradients an extremely fine spatial resolution is needed, therefore we ran all the simulations presented in Fig. 7 with  $\Delta y = 10^{-4}$  except for the two biggest values of  $a$  and the two smallest values of  $D$  where even  $\Delta y = 10^{-5}$  is needed. If we do not have such fine spatial resolution the axon continues its growth far past its steady-state length. We consider that to be an unrealistic solution.

Fig. 9 shows that the axon length is sensitive to changes in the degradation rate  $g$ .

Recall the relation between the parameters  $l_c$ ,  $r_g$ , and  $c_c^\infty$  as well as the relation between  $\tilde{r}_g$ ,  $r_g$ , and  $c_c^\infty$  defined by the formulas (3)–(5). In the simulations presented in Fig. 10(a)



**Fig. 9** Axon length  $l(t)$  for varying degradation rate  $g$ .

the parameter  $l_c$  is varied without changing  $c_c^\infty$  and  $r_g$  (this means that  $\tilde{s}_g \kappa$  is varied with  $l_c$  to keep  $s_g$  constant). From Diehl et al (2014, Fig. 8(b)) we know that the steady-state length is insensitive to changes in the growth cone length parameter  $l_c$ . This is in accordance with Fig. 10(a) where we also can see that the growth speed is sensitive to these variations; the axon grows close to its steady-state length only for the two smallest values of  $l_c$ .

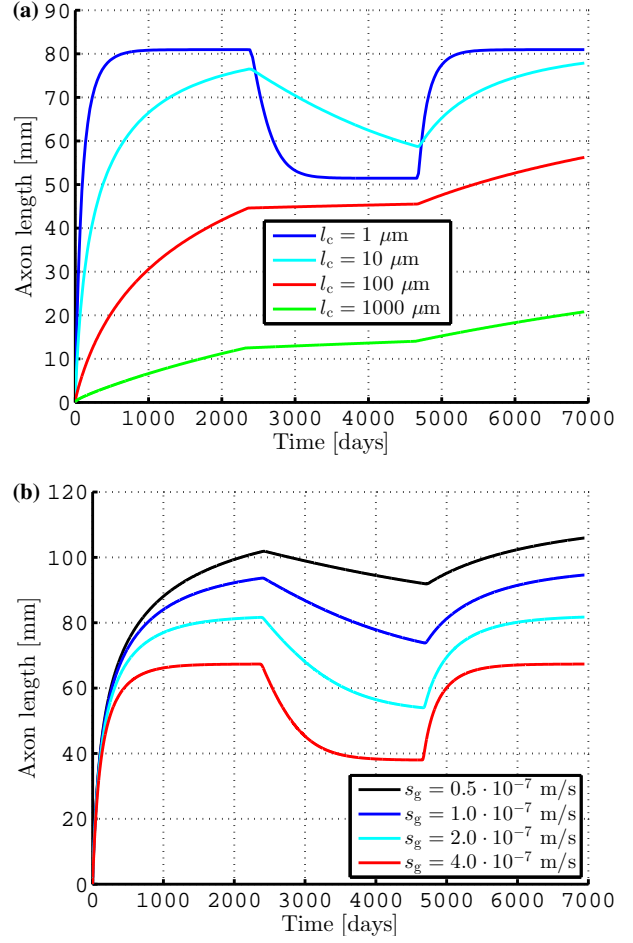
In Fig. 10(b) we instead keep  $l_c$  constant and let the maximum speed of shrinkage  $s_g$  vary. This affects the model (1) through the relation  $c_c^\infty = s_g/r_g$  (where  $r_g$  is kept constant). We can observe that the model indeed reacts faster to changes in soma concentration for larger values of  $s_g$ . Further, for values of  $s_g \gtrsim 4.3 \cdot 10^{-7}$  m/s we have  $c_s(t)/c_c^\infty < 1$  even for small values of  $t$ . This means that we cannot expect any outgrowth for these values of  $s_g$ , cf. Diehl et al (2014, Thm 4.1 and Fig. 11).

In Fig. 11 the effects of changing the polymerization rates  $\tilde{r}_g$  and  $r_g$  are investigated while keeping  $s_g$  constant. We can observe that the steady-state length is sensitive to  $r_g$  but unaffected by changes in  $\tilde{r}_g$  (as was already known from Diehl et al (2014)). The opposite dependency seems to apply for the profile of the axon growth with  $\tilde{r}_g$  affecting the shape while  $r_g$  does not. Also note that for  $r_g \lesssim 8.9 \cdot 10^{-6}$  m<sup>4</sup>/(mol s) we have  $c_s(t)/c_c^\infty < 1$  so we cannot expect any outgrowths for these values of  $r_g$ .

For the simulations presented in Fig. 12 different functions defining the soma concentration are used. To this end, we let  $c_s(t)$  be defined by (26) and set

$$\begin{aligned} c_s^{[1]}(t) &:= 0.5 \cdot c_s(t), & c_s^{[2]}(t) &:= 0.75 \cdot c_s(t), \\ c_s^{[3]}(t) &:= c_s(t), & c_s^{[4]}(t) &:= 1.25 \cdot c_s(t). \end{aligned}$$

For these particular simulations, to get continuity at  $(x, t) = (0, 0)$ , also the initial condition must be rescaled. That is, for  $k = 1, 2, 3, 4$ , we use the constant initial data  $c(x, 0) = c_s^{[k]}(0)$ .



**Fig. 10** Axon length  $l(t)$  for: (a) varying  $l_c$  (with  $s_g$  constant at its nominal value, cf. Table 1) and (b)  $s_g$  and  $c_c^\infty$  varying together according to  $c_c^\infty = s_g/r_g$  (with  $l_c$  and  $r_g$  nominal).

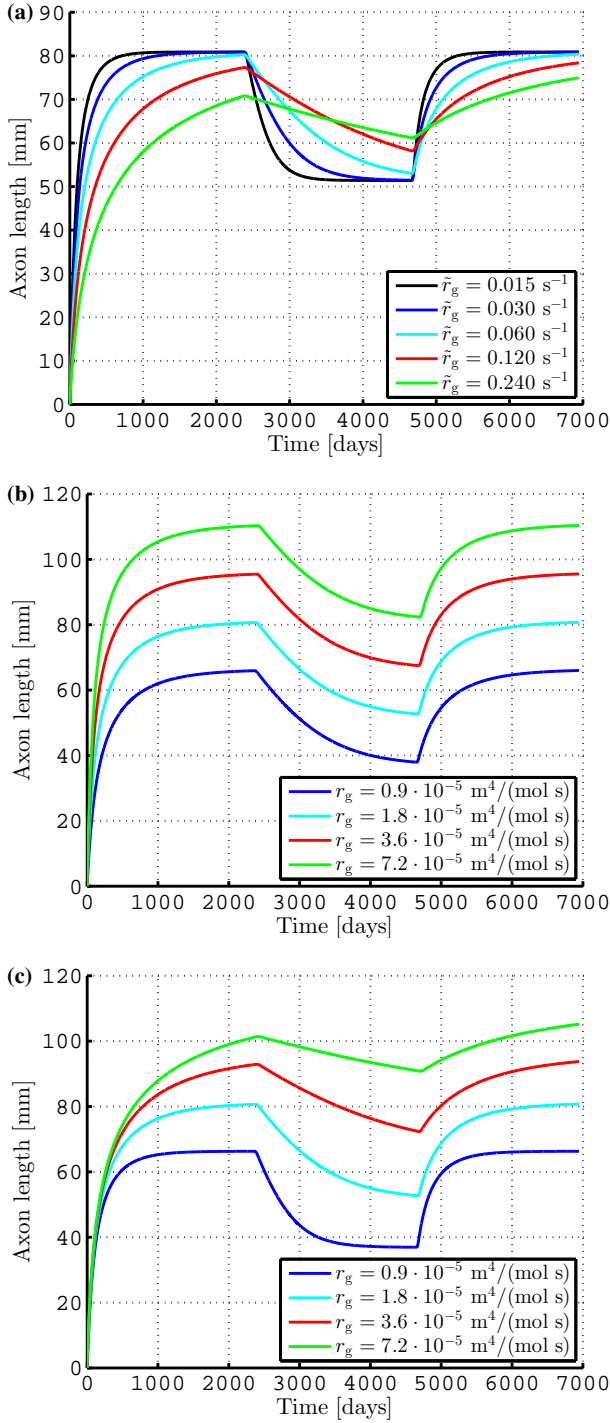
Fig. 11 shows that larger soma concentrations give longer axon lengths without changing the growth profile.

Finally, we keep all parameters at their nominal values and run simulations for a range of different values of  $l^0$  between  $10^{-6}$  μm and  $10^3$  μm. Since this variation has no visible effect on the solution on the considered time scale we do not include a plot. We also note that the initial length has marginal effect on the CPU time of the simulation; for the shortest  $l^0$  the CPU time is 57 s whereas for the longest initial length it is 55 s.

## 6 Conclusions

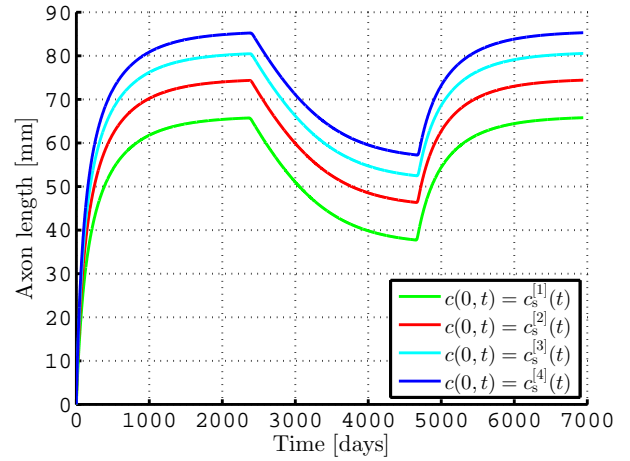
The model of axonal growth, consisting of two ODEs coupled to one PDE defined on an interval, which length is one unknown of the problem, has shown to be a challenge to simulate dynamically. After a spatial transformation of the moving boundary, it is still not straightforward to apply any numerical method. Standard methods cannot be used to simulate the entire axonal growth, from a very small length to





**Fig. 11** Axon length  $l(t)$  for: (a) varying  $\tilde{r}_g$  (with  $r_g$ ,  $s_g$ , and  $c_c^\infty$  nominal), (b)  $r_g$  and  $c_c^\infty$  varying together according to (5) (with  $\tilde{r}_g$  and  $s_g$  nominal), and (c)  $r_g$ ,  $\tilde{r}_g$ , and  $c_c^\infty$  varying together according to (4) and (5) (with  $s_g$  nominal).

the final one several magnitudes larger, in reasonable CPU time. Furthermore, the low accuracy of first-order methods can easily simulate that the axon grows beyond its steady state, obtained as an exact solution of the model equations; hence, such numerical solutions are not reliable. To obtain



**Fig. 12** Axon length  $l(t)$  for different functions defining the left-hand side boundary condition (the soma concentration).

reliable and efficient (fast and accurate) numerical solutions, it was necessary to use both an additional time transformation of the equations and an application of the Peaceman-Rachford splitting scheme.

With the efficient numerical scheme presented, investigations of the biological and physical parameters in the model were performed. These investigations complements those for the steady-state solution of the model made in our previous publication Diehl et al (2014). The following conclusions can be made for the dynamic behaviour:

- The axon grows very fast initially for a broad range of parameter values. As is also known from experiments, the active transport is the dominant driving force over diffusion for the growth velocity and the final length of the axon. These findings are in qualitative agreement with those presented by Graham et al (2006, Figure 4).
- The concentration of free tubulin in the growth cone approaches quickly the steady-state concentration after any change in the driving soma concentration  $c_s(t)$ .
- The concentration profile of tubulin along the axon, during growth, is similar to the characteristic steady-state profile, which is generally decreasing with the distance from the soma to a minimum value close to the growth cone and then increasing rapidly just before the cone.
- If the size of the growth cone is increased from  $1 \mu\text{m}$  to  $10 \mu\text{m}$ , the growth velocity is decreased but the final axon length is about the same.
- The numerical scheme gives the possibility for a careful examination of the influence of the variations of the parameters related to the (de)polymerization on the dynamic solution. For example, the polymerization rate coefficient  $\tilde{r}_g$  does not influence the final axon length (known from Diehl et al (2014)), but it has a great influence on the growth velocity of the axon; see Fig. 11(a).
- In Sec. 3.2 we have described time-scaling with respect to advection  $a$  (9) and with respect to  $D$  (13). The scaling

with  $D$  performs better when the axon is short whereas the one with  $a$  performs better when the axon is long, see Fig. 6. This is in agreement with the expected behaviour of advection and diffusion at different length scales. Note that, switching between the scalings at some intermediate axon length could give a scheme with improved performance. This could be of importance if our model were to be expanded to a larger one that would be more expensive to simulate. For the studies carried out in this article, however, the usage of only scaling with  $a$  has shown to be sufficient.

## Acknowledgements

The authors thank the reviewers for valuable suggestions, in particular, the idea to investigate the scaling with respect to diffusion.

## A Proof of Lemma 1

We begin by noting that any positive definite matrix is invertible and that any square matrix is positive definite when its symmetric part has only positive eigenvalues. That is, the linear system of equations (23) has a unique solution when the eigenvalues of

$$\mathbf{S}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) := \frac{\left(\mathbf{I} - \frac{\Delta\tau}{2}\mathbf{A}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y})\right) + \left(\mathbf{I} - \frac{\Delta\tau}{2}\mathbf{A}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y})\right)^T}{2}$$

are all positive. The entries of the main diagonal, respectively the super and sub diagonals are given as follows

$$\begin{aligned} s_{j,j}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) &= 1 + \frac{\Delta\tau}{2} \left( \frac{2D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} + \frac{g}{a} L^{n+\frac{1}{2}} \right), \quad j = 1, \dots, M-1, \\ s_{j,j+1}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) &= s_{j+1,j}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) \\ &= -\frac{\Delta\tau}{2} \left( \frac{r_g}{4a} (C_c^{n+\frac{1}{2}} - c_c^\infty) + \frac{D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} \right), \quad j = 1, \dots, M-2. \end{aligned}$$

Thus,  $\mathbf{S}(\mathbf{U}^{n+1/2}, \mathbf{y})$  is a symmetric, tridiagonal Toeplitz matrix meaning that the eigenvalues are given by the following formula

$$\lambda_k = s_{j,j}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) + 2s_{j,j+1}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) \cos(k\pi\Delta y), \quad k = 1, \dots, M-1.$$

Then, any  $\lambda_k$  fulfils the following inequality

$$\begin{aligned} \lambda_k &\geq s_{j,j}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) - |2s_{j,j+1}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) \cos(k\pi\Delta y)| \\ &\geq s_{j,j}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y}) - |2s_{j,j+1}(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{y})|. \end{aligned}$$

By plugging in the entries and using the triangle inequality we get

$$\begin{aligned} \lambda_k &\geq 1 + \frac{\Delta\tau}{2} \left( \frac{2D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} + \frac{g}{a} L^{n+\frac{1}{2}} \right) \\ &\quad - \Delta\tau \left| \frac{r_g}{4a} (C_c^{n+\frac{1}{2}} - c_c^\infty) + \frac{D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} \right| \\ &\geq 1 + \frac{\Delta\tau}{2} \left( \frac{2D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} + \frac{g}{a} L^{n+\frac{1}{2}} \right) \\ &\quad - \Delta\tau \left( \left| \frac{r_g}{4a} (C_c^{n+\frac{1}{2}} - c_c^\infty) \right| + \left| \frac{D}{a(\Delta y)^2} \frac{1}{L^{n+\frac{1}{2}}} \right| \right) \\ &= 1 + \Delta\tau \left( \frac{g}{2a} L^{n+\frac{1}{2}} - \frac{r_g}{4a} |C_c^{n+\frac{1}{2}} - c_c^\infty| \right) \\ &\geq 1 - \Delta\tau \frac{r_g}{4a} |C_c^{n+\frac{1}{2}} - c_c^\infty| \geq 1 - \Delta\tau \frac{r_g}{4a} \gamma, \end{aligned}$$

where we have used the definition of  $\gamma$  given by (24). We conclude that the eigenvalues of  $\mathbf{S}(\mathbf{U}^{n+1/2}, \mathbf{y})$  are positive when

$$\Delta\tau < \frac{4a}{r_g} \cdot \frac{1}{\gamma},$$

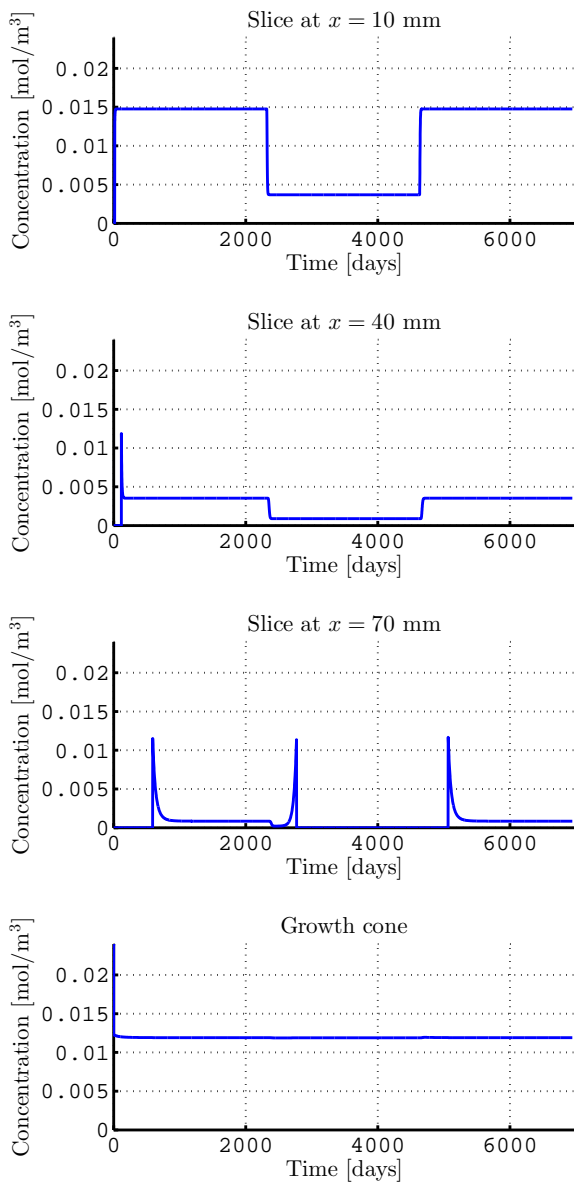
and therefore, for these values of  $\Delta\tau$ , the system (23) has a unique solution.

## B Two-dimensional slices of Figures 2(b) and 8

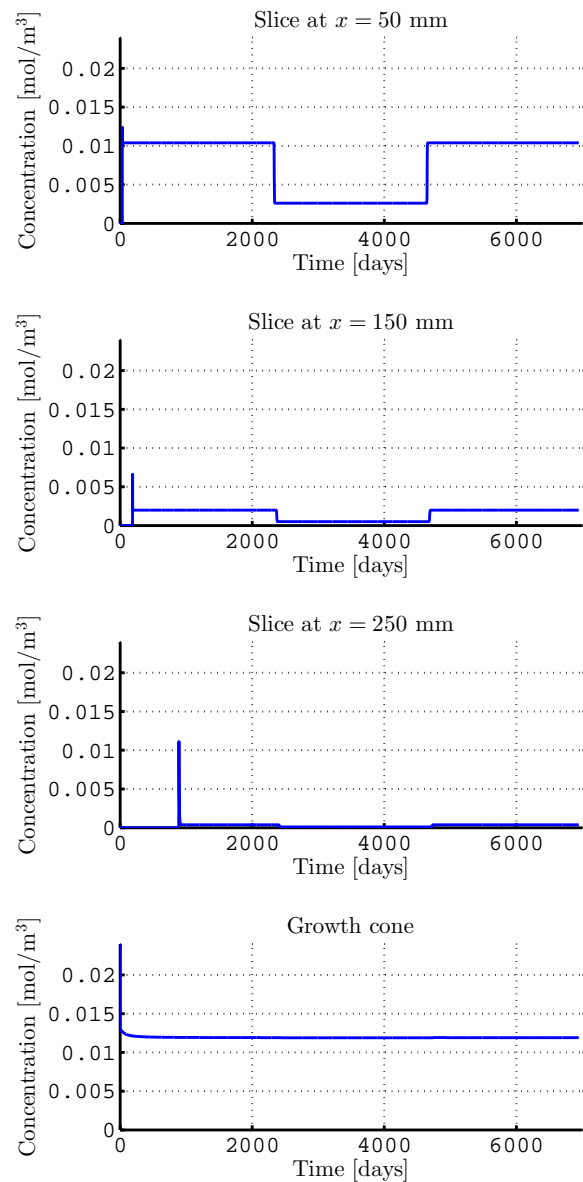
We complement the three-dimensional plots (Figs. 2(b) and 8) with two-dimensional slices at different  $x$  and  $t$  values. Recall that in Fig. 2(b) the tubulin concentration along the axon is plotted for nominal values on the biological and physical parameters. For Fig. 8 a three times larger advection velocity  $a$  is used. The slices of 2(b) and 8 are presented next to each other in Figs. 13–16 for easy comparison.

## References

- Diehl S, Henningsson E, Heyden A, Perna S (2014) A one-dimensional moving-boundary model for tubulin-driven axonal growth. *Journal of Theoretical Biology* 358:194–207
- Douglas J (1955) On the numerical integration of  $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$  by implicit methods. *Journal of the Society for Industrial and Applied Mathematics* 3(1):42–65
- García JA, Peña JM, McHugh S, Jérusalem A (2012) A model of the spatially dependent mechanical properties of the axon during its growth. *CMES – Computer Modeling in Engineering and Sciences* 87(5):411–432
- Graham BP, van Ooyen A (2006) Mathematical modelling and numerical simulation of the morphological development of neurons. *BMC Neuroscience* 7(Suppl. 1)
- Graham BP, Lauchlan K, McLean DR (2006) Dynamics of outgrowth in a continuum model of neurite elongation. *Journal of Computational Neuroscience* 20(1):43–60
- Hansen E, Henningsson E (2013) A convergence analysis of the Peaceman–Rachford scheme for semilinear evolution equations. *SIAM Journal on Numerical Analysis* 51(4):1900–1910
- Hundsdoerfer W, Verwer J (2003) *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Series in Computational Mathematics, vol 33. Springer, New York
- Kiddie G, McLean D, Ooyen AV, Graham B (2005) Biologically plausible models of neurite outgrowth. In: van Pelt J, Kamermans M, Levelt CN, van Ooyen A, Ramakers GJA, Roelfsema PR (eds) *Development, Dynamics and Pathology of Neuronal Networks: from Molecules to Functional Circuits*, Progress in Brain Research, vol 147, Elsevier, pp 67–80



**Fig. 13** Slices of Fig. 2(b) at three different  $x$  values and at the growth cone. Note that the spatial position of the latter changes with time. The spikes in the second and third subfigures correspond to times where the respective slice is close to the cone. Further, note that the concentration in the growth cone is largely unaffected by the soma concentration, instead a higher value in the latter results in a longer axon.

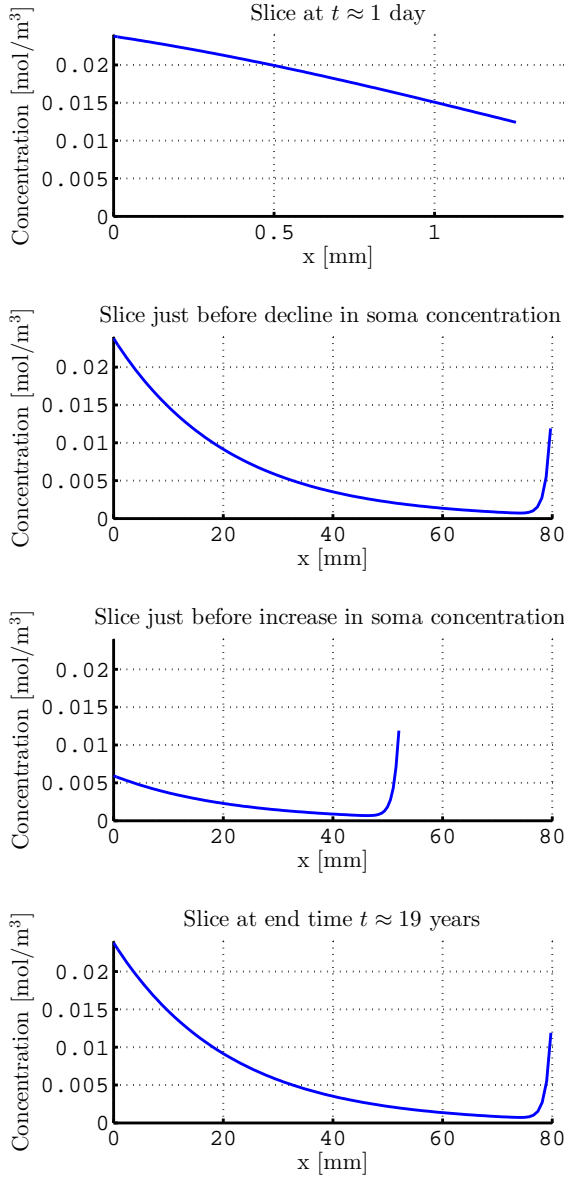


**Fig. 14** Slices of Fig. 8 at three different  $x$  values and at the growth cone. As the three times larger advection velocity  $a$  gives a far longer axon, different  $x$  values are chosen compared to those in Fig. 13. Note how the higher advection velocity gives smaller concentration values in the inner of the axons; the tubulin is concentrated close to the soma and the growth cone.

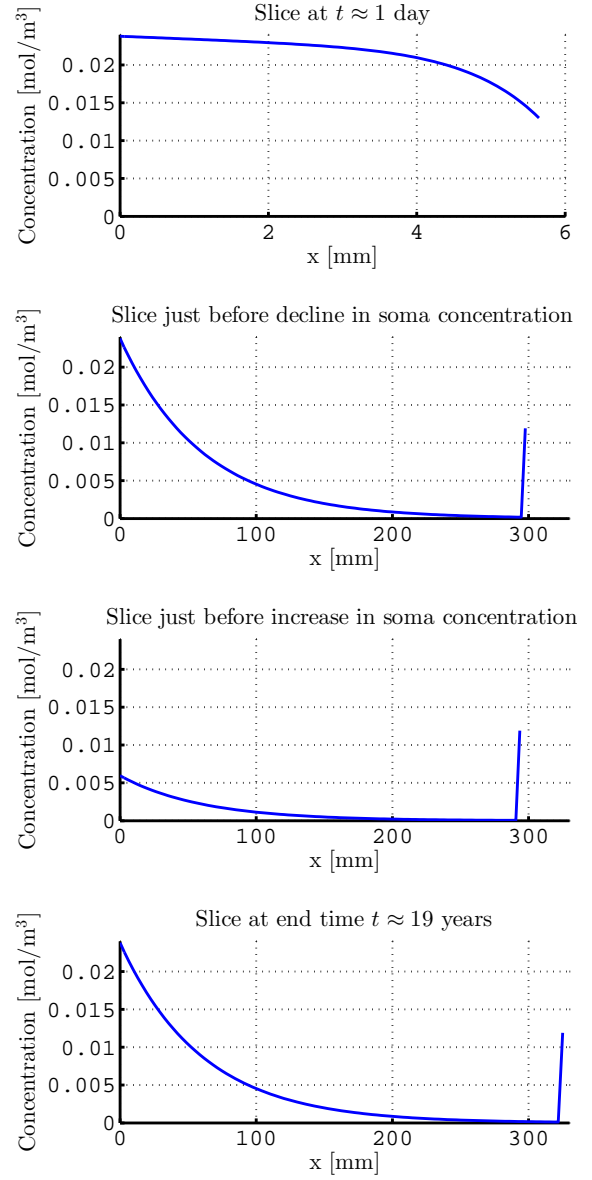
- McLean DR, Graham BP (2004) Mathematical formulation and analysis of a continuum model for tubulin-driven neurite elongation. *Proceedings Royal Society A: Mathematical, Physical and Engineering Sciences* 460(2048):2437–2456
- McLean DR, Graham BP (2006) Stability in a mathematical model of neurite elongation. *Mathematical Medicine and Biology – A Journal of the IMA* 23(2):101–117
- McLean DR, van Ooyen A, Graham BP (2004) Continuum model for tubulin-driven neurite elongation. *Neurocomputing* 58–60:511–516
- Miller KE, Heidemann SR (2008) What is slow axonal transport? *Experimental Cell Research* 314(10):1981–1990
- Peaceman DW, Rachford HH (1955) The numerical solution of parabolic and elliptic differential equations. *Journal of the Society*

- for Industrial and Applied Mathematics* 3(1):28–41
- Sadegh Zadeh K, Shah SB (2010) Mathematical modeling and parameter estimation of axonal cargo transport. *Journal of Computational Neuroscience* 28(3):495–507
- Smith DA, Simmons RM (2001) Models of motor-assisted transport of intracellular particles. *Biophysical Journal* 80(1):45–68
- Suter DM, Miller KE (2011) The emerging role of forces in axonal elongation. *Progress in Neurobiology* 94(2):91–101
- van Ooyen A (2011) Using theoretical models to analyse neural development. *Nature Reviews Neuroscience* 12(6):311–326
- Walker RA, O'Brien ET, Pryer NK, Soboeiro MF, Voter WA, Erickson HP, Salmon ED (1988) Dynamic instability of individual microtubules analyzed by video light microscopy: rate constants and





**Fig. 15** Slices of Fig. 2(b) at four different  $t$  values. See also (26) for the times of decline and increase in soma concentration. Note the characteristic profile of the concentration along the axon, cf. Diehl et al (2014, Sec. 4).



**Fig. 16** Slices of Fig. 8 at four different  $t$  values. Note the increased length of the axon as an effect of the three times larger advection velocity  $a$ . Note also the sharper gradient close to the growth cone, cf. Diehl et al (2014, Sec. 4).